Setting Database Options[1]*

February 10, 2000

* Because this topic is not covered by the current PMO funded SPS Helpdesk Agreement, this document has been provided to help you resolve this issue. If you still need assistance after reviewing this document, please contact a representative from your Customer Support Team.

[1] *Sybase Adaptive Server Enterprise System Administration Guide*. Emeryville, CA: Sybase, Inc., Sept 1997: pp 16-1 to 16-10.

Table of Contents

1. Introduction

Database options control many different aspects of database behavior, such as:

- The behavior of transactions
- Defaults for table columns
- Restrictions to user access
- Performance of recovery and bcp operations
- Log behavior

System Administrators and Database Owners can use database options to configure the settings for an entire database.

2. Database Option Descriptions

2.1 Required Options (for PD² version 4.1a and 4.1b)

2.1.1 Trunc log on chkpt

The "trunc log on chkpt" option means that the transaction log is truncated (committed transactions are removed) when the checkpoint checking process occurs (usually more than once per minute) if 50 or more rows have been written to the log. The log is not truncated if less than 50 rows were written to the log, or if the Database Owner runs the checkpoint command manually.

If this option is off (the original default condition) and the transaction log is never dumped, the transaction log continues to grow and you may run out of space in your transaction log causing the system to become hung up until the log is dumped.

When the "trunc log on chkpt" option is on, you cannot back up the transaction log because changes to your data are not recoverable from transaction log dumps. In this situation, issuing the "dump transaction…to" command produces an error message instructing you to use dump database instead.

Note: With the advent of 4.1c the "truncate log on checkpoint" option can be turned off to allow for up-to-the-minute recovery of the database. Detailed instructions on how to dump the transaction log can be found in the paper entitled "Sybase Server Maintenance Schedule".

2.1.2 Select into/bulkcopy

The select into/bulkcopy option must be set to "on" in order to perform operations that do not keep a complete record of the transaction in the log such as:

- Auto Clause Selection
- Running the Clause Database Installer
- Upgrading the Database

SQL Server performs minimal logging for these commands, recording only page allocations and deallocations, but not the actual changes that are made on the data pages.

You do not have to set the select into/bulkcopy option to on in order to select into a temporary table, since tempdb is never recovered. The option does not need to be set in order to run bcp on a table that has indexes, because inserts are logged.

After you have run a select into command or performed a bulk copy in a database, you will not be able to perform a regular transaction log dump. Once you have made minimally logged changes to your database, you must perform a dump database, since changes are not recoverable from transaction logs.

Just setting the select into/bulkcopy option does not block log dumping, but making minimally logged changes to data does block the use of a regular dump transaction. However, you can still use dump transaction...with no_log and dump transaction...with truncate_only.

By default, the select into/bulkcopy option is turned off in newly created databases. To change the default situation, turn this option on in the model database.

2.2 Remaining Options

Note: Please consult the Help Desk prior to changing any of the following options.

2.2.1 Abort tran on log full

Abort tran on log full determines the fate of a transaction that is running when the last-chance threshold is crossed. The default value is false, meaning that the transaction is suspended and is awakened only when space has been freed. If you change the setting to true, all user queries that need to write to the transaction log are killed until space in the log has been freed.

2.2.2 Allow nulls by default

Setting allow nulls by default to true changes the default null type of a column from not null to null, in compliance with the SQL standard. The Transact-SQL default value for a column is not null, meaning that null values are not allowed in a column unless null is specified in the column definition.

2.2.3 Auto identity

While the auto identity option is true, a 10-digit IDENTITY column is defined in each new table that is created without specifying either a primary key, a unique constraint, or an IDENTITY column. The column is not visible when you select all columns with the select * statement. To retrieve it, you must explicitly mention the column name, SYB_IDENTITY_COL, in the select list.

To set the precision of the automatic IDENTITY column, use the size of auto identity configuration parameter.

2.2.4 dbo use only

While the dbo use only option is set to "on" (true), only the Database Owner can use the database.

2.2.5 ddl in tran

Setting the ddl in tran option to true allows data definition commands to be used inside a user-defined transaction.

Data definition statements must lock system tables for the duration of a transaction, which can result in performance problems. Use them only in short transactions.

2.2.6 Identity in nonunique index

The identity in nonunique index option automatically includes an IDENTITY column in a table's index keys so that all indexes created on the table are unique. This database option makes logically non-unique indexes internally unique, and allows those indexes to be used to process updatable cursors and isolation level 0 reads.

The table must already have an IDENTITY column for the identity in nonunique index option to work, either from a create table statement or by setting the auto identity database option to true before creating the table.

Use identity in nonunique index if you plan to use cursors and isolation level 0 reads on tables that have non-unique indexes. A unique index ensures that the cursor will be positioned at the correct row the next time a fetch is performed on that cursor.

2.2.7 No chkpt on recovery

The no chkpt on recovery option is set to true (on) when an up-to-date copy of a database is kept. In these situations, there is a "primary" and a "secondary" database. Initially, the primary database is dumped and loaded into the secondary database. Then, at intervals, the transaction log of the primary database is automatically dumped and loaded into the secondary database.

If this option is set to "off" (false) the default condition a checkpoint record is added to the database after it is recovered due to restarting SQL Server. This checkpoint, which ensures that the recovery mechanism is not rerun unnecessarily, changes the sequence number on the database. If the sequence number on the secondary database has been changed, a subsequent dump of the transaction log from the primary database cannot be loaded into it.

Turning on this option for the secondary database causes it not to get a checkpoint from the recovery process so that subsequent transaction log dumps from the primary database can be loaded into it.

2.2.8 No free space acctg

The no free space acctg option suppresses free space accounting and execution of threshold actions for the non-log segments. This speeds up recovery time because the free-space counts will not be recomputed for those segments. It disables updating the rows-per-page value stored for each table, so system procedures that estimate space usage may report inaccurate values.

2.2.9 Read only

The read only option means that users can retrieve data from the database, but cannot modify anything.

2.2.10 Single user

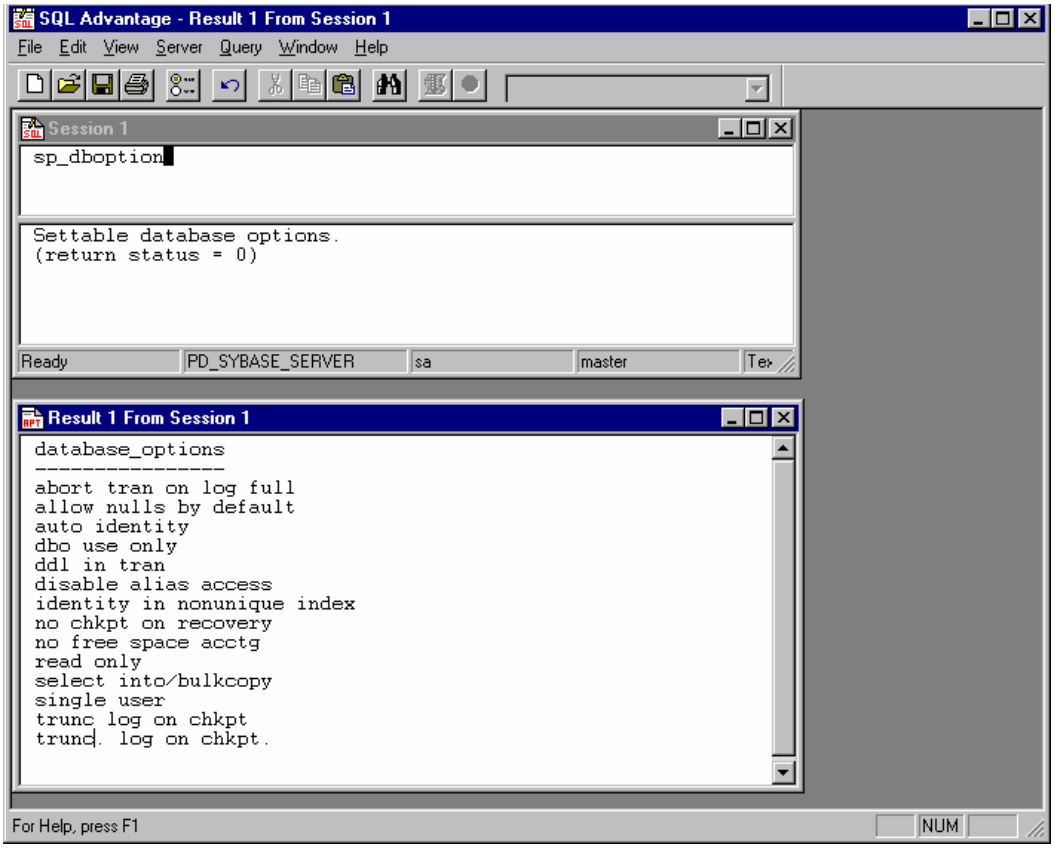
When single user is set to true, only one user at a time can access the database.

3. Setting Database Options

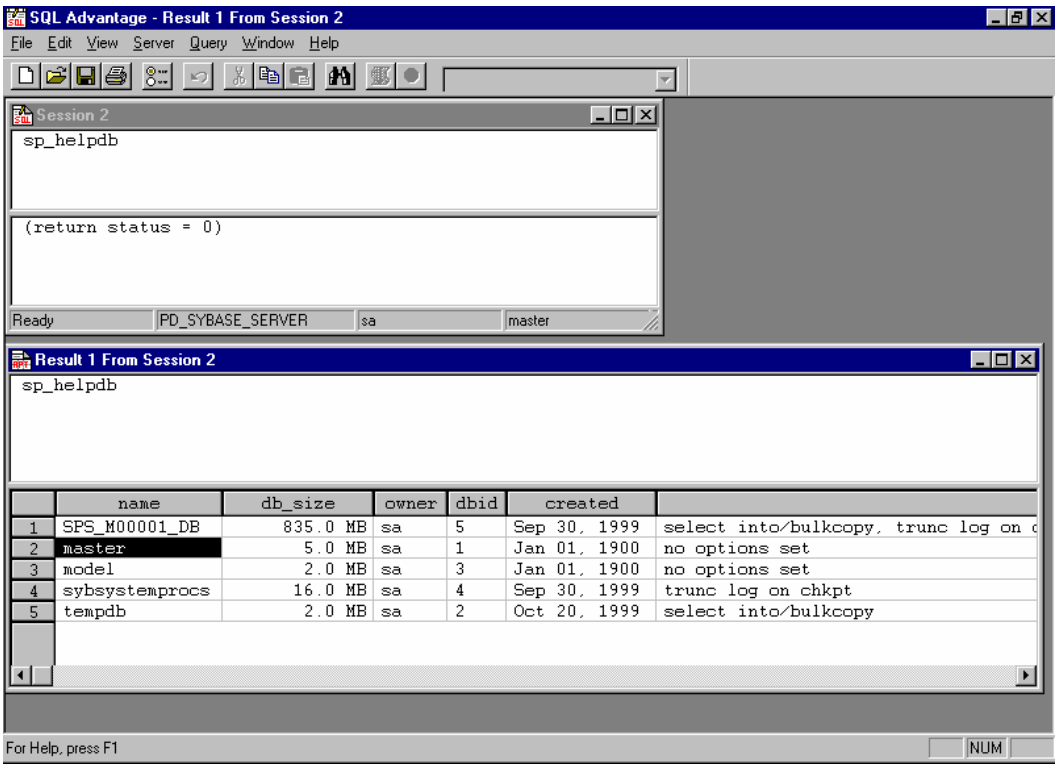
3.1 Using WISQL or SQL Advantage

3.1.1 Viewing Options

All users with access to the master database in WISQL or SQL Advantage can execute the sp_dboption procedure with no parameters to display a list of the database options. The report from sp_dboption looks like this.



For a report on which options have been set in a particular database, execute the system procedure sp_helpdb.



3.1.2 Setting Options

Database options can be changed for any database except for the master database. To change a database option in a PD² database you must execute sp_dboption from the master database.

The syntax for sp_dboption is as follows:

```
sp_dboption <dbname>, "<optname>", {true|false}
```

<dbname> = the name of the database for which you're setting the option.
<optname> = the name of the option that you want to set.
<true|false> = the choice of setting. True = on. False = off.

After setting the options run the *checkpoint* command in your database for the changes to take effect.

```
use <dbname>
go

checkpoint
go
```

<dbname> = the name of the database for which you've changed the option.